

21/02/2025

Projet LPIC-101

Rapport de projet



PERREIN Alcide - NANDILLON Maxence

3ESGI, Nantes

Table des matières

[1. Introduction 1](#_Toc190450626)

[1.1 Contexte du projet 1](#_Toc190450627)

[1.2 Objectifs du projet 2](#_Toc190450628)

[1.3 Environnement de travail 2](#_Toc190450629)

[2. Gestion de projet 2](#_Toc190450630)

[2.2 Répartition des tâches 2](#_Toc190450631)

[2.3 Outils de gestion de projet 3](#_Toc190450632)

[3. Développement technique 3](#_Toc190450633)

[3.1 Architecture du système 3](#_Toc190450634)

[3.2 Fonctionnalités implémentées 4](#_Toc190450635)

[3.3 Outils et technologies utilisés 4](#_Toc190450636)

[4. Difficultés rencontrées 4](#_Toc190450637)

[4.1 Problèmes techniques 4](#_Toc190450638)

[4.2 Solutions apportées 4](#_Toc190450639)

[5. Conclusion 5](#_Toc190450640)

[5.1 Bilan du projet 5](#_Toc190450641)

[5.2 Perspectives d’amélioration 5](#_Toc190450642)

[6. Annexes 5](#_Toc190450643)

[6.1 Guide d’utilisation 5](#_Toc190450644)

[6.2 Liste des outils et services utilisés 5](#_Toc190450645)

## 1. Introduction

### 1.1 Contexte du projet

Le projet LPIC-101 a été initié dans le cadre de notre formation à l’ESGI en troisième année, option Systèmes, Réseaux et Cloud (SRC). Ce projet a été conçu par notre professeur de Linux afin de nous préparer à la certification LPIC-101, une certification reconnue dans le domaine de l’administration système Linux. L’objectif principal était de créer une application en ligne de commande sous Debian pour gérer les tâches d’administration système dans une petite entreprise.

### 1.2 Objectifs du projet

L’objectif du projet était de développer plusieurs scripts Bash pour offrir un menu interactif permettant de :

* Gérer les utilisateurs et les groupes.
* Automatiser des tâches via Cron.
* Surveiller l’espace disque et les processus.
* Sauvegarder des dossiers spécifiques.
* Configurer la journalisation système.

Ces scripts devaient être conçus pour être utilisés par les agents RH, les employés et l’équipe IT de l’entreprise.

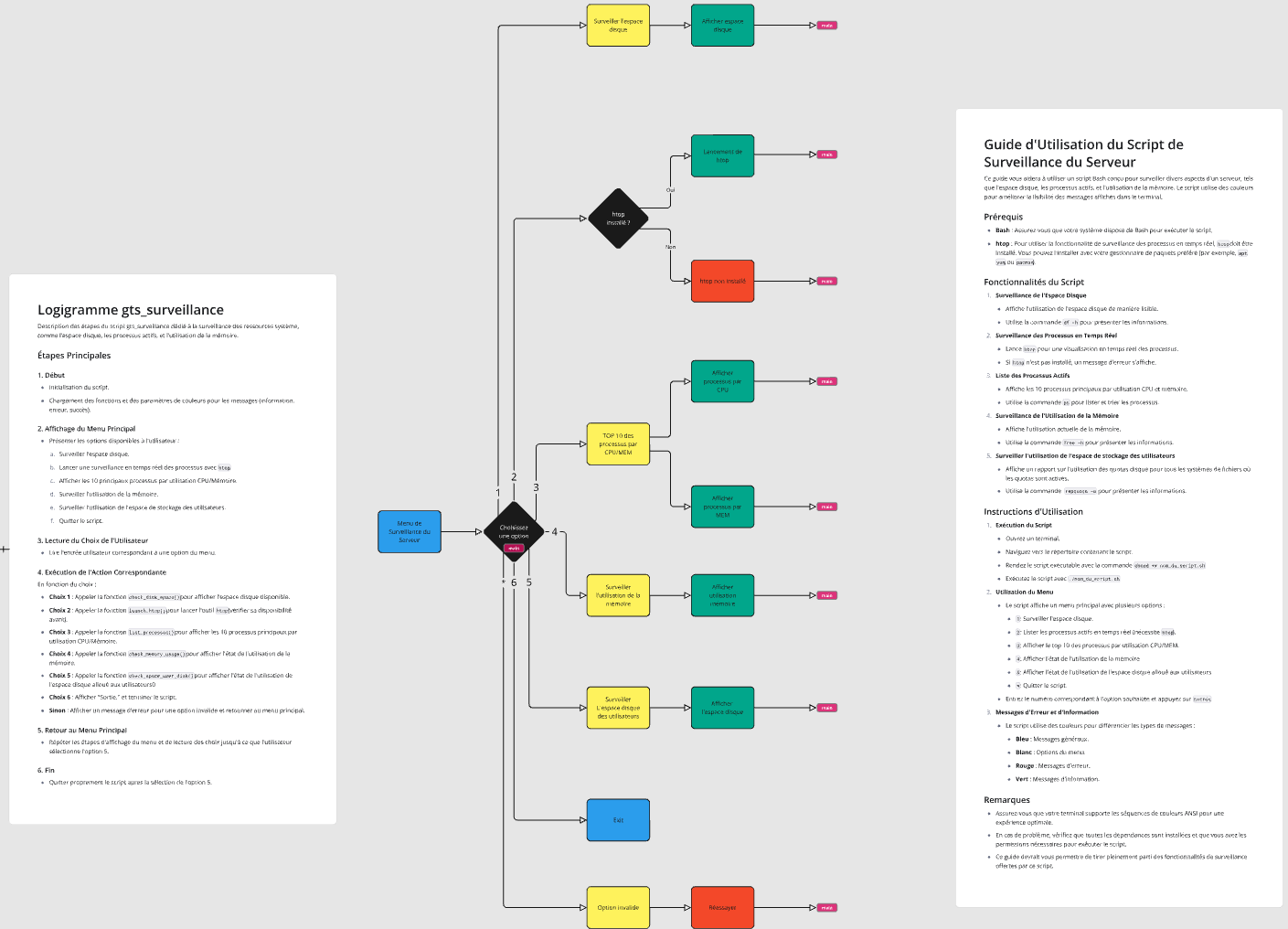
### 1.3 Environnement de travail

Pour réaliser ce projet, nous étions deux étudiants en troisième année à l’ESGI, en spécialité SRC, préparant le diplôme de chef de projet logiciel et réseaux. Nous avons travaillé en collaboration, en utilisant des outils de gestion de projet et des technologies modernes pour garantir une réalisation efficace et professionnelle.

z

## 2. Gestion de projet

### 2.2 Répartition des tâches

La répartition des tâches a été réalisée en fonction des compétences de chacun. Nous avons utilisé **Miro** pour créer des logigrammes, des diagrammes UML et des documents d’utilisation. Cela nous a permis de visualiser clairement les responsabilités de chacun et de suivre l’avancement du projet.

### 2.3 Outils de gestion de projet

* Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

  Description générée automatiquement**Git et GitHub** : Nous avons utilisé Git pour le versioning du code et GitHub pour le stockage et la collaboration. Nous avons également mis en place des **Hooks**, des **GitHub Actions** et des pipelines CI/CD pour automatiser les tests et les déploiements.
* **SonarQube** : Pour la détection de vulnérabilités et la qualité du code.
* **Miro** : Pour la création de diagrammes et la planification des tâches.
* **Vagrant, VirtualBox** **et** **VMware**: Pour la création d’environnements virtuels reproductibles.

## 3. Développement technique

### 3.1 Architecture du système

Le système est basé sur une architecture modulaire, où chaque fonctionnalité est gérée par un script Bash distinct. Les scripts sont conçus pour être exécutés par des groupes d’utilisateurs spécifiques, garantissant ainsi une sécurité et une gestion des droits adaptées. !!!!!! #NOTE POUR MAXENCE TAB DROITS /PERMS UTILISATEURS !!!!!

### 3.2 Fonctionnalités implémentées

* **Gestion des utilisateurs et des groupes** : Création, suppression et affectation des utilisateurs aux groupes.
* **Automatisation avec Cron** : Planification et gestion des tâches cron.
* **Surveillance système** : Surveillance de l’espace disque, des processus et de la mémoire.
* **Sauvegarde de dossiers** : Sauvegarde manuelle et automatique des dossiers.
* Une image contenant texte, Police, capture d’écran

  Description générée automatiquement**Configuration de la journalisation** : Configuration de rsyslog pour la centralisation des logs.

### 3.3 Outils et technologies utilisés

* **Bash** : Pour la création des scripts.
* **Vagrant** : Pour la création et la gestion des machines virtuelles.
* **doas** : Alternative à sudo pour la gestion des permissions.
* **chmod 2750** : Pour la gestion des permissions spécifiques.
* **Codviz** : Pour la création d’organigrammes des scripts.

## 4. Difficultés rencontrées

### 4.1 Problèmes techniques

* **Configuration de GRUB** : La personnalisation de GRUB pour un démarrage sécurisé a été complexe, notamment en raison des spécificités du système Debian.
* **Gestion des permissions** : La mise en place de permissions spécifiques sans utiliser le fichier sudoers a nécessité des recherches approfondies.
* **Intégration de Cron** : L’automatisation des tâches via Cron a posé des problèmes de synchronisation et de gestion des erreurs.

### 4.2 Solutions apportées

* Pour GRUB, nous avons consulté la documentation officielle et testé plusieurs configurations avant de trouver la solution optimale.
* Pour les permissions, nous avons utilisé doas et chmod 2750 pour contourner les limitations de sudoers.
* Pour Cron, nous avons implémenté des scripts de vérification pour garantir que les tâches s’exécutent correctement.

## 5. Conclusion

### 5.1 Bilan du projet

Le projet LPIC-101 a été une expérience enrichissante qui nous a permis de mettre en pratique les connaissances acquises en cours. Nous avons réussi à développer une application fonctionnelle répondant aux besoins d’une petite entreprise en matière d’administration système.

### 5.2 Perspectives d’amélioration

* **Interface graphique** : Ajouter une interface graphique pour faciliter l’utilisation par des utilisateurs non techniques.
* **Automatisation avancée** : Intégrer des outils d’automatisation plus poussés comme Ansible.
* **Monitoring en temps réel** : Ajouter des fonctionnalités de monitoring en temps réel pour une meilleure surveillance du système.

## 6. Annexes

### 6.1 Guide d’utilisation

Un guide d’utilisation détaillé est disponible sur le dépôt GitHub du projet : <https://github.com/NANDILLONMaxence/Projet-LPIC-101>

### 6.2 Liste des outils et services utilisés

* **Git et GitHub** : Versioning et collaboration.
* **Vagrant et VirtualBox** : Environnements virtuels.
* **Miro** : Diagrammes et planification.
* **SonarQube** **et Github Action** : Qualité du code.
* **Codviz** : Organigrammes des scripts.